A MIMD IMPLEMENTATION OF A PARALLEL EULER SOLVER FOR UNSTRUCTURED GRIDS

V. Venkatakrishnan, Horst D. Simon,
and Timothy J. Barth

NAS Applied Research Branch
NASA Ames Research Center
Moffett Field, CA 94035
simon@orville.nas.nasa.gov

Fifth SIAM Conference on Parallel Processing
Houston, TX, March 25 - 27, 1991

OUTLINE

- Introduction to the problem
- Description of the MIMD parallel Euler solver
- Algorithms for the partitioning problem
- Optimization issues on the iPSC/860
- Numerical results

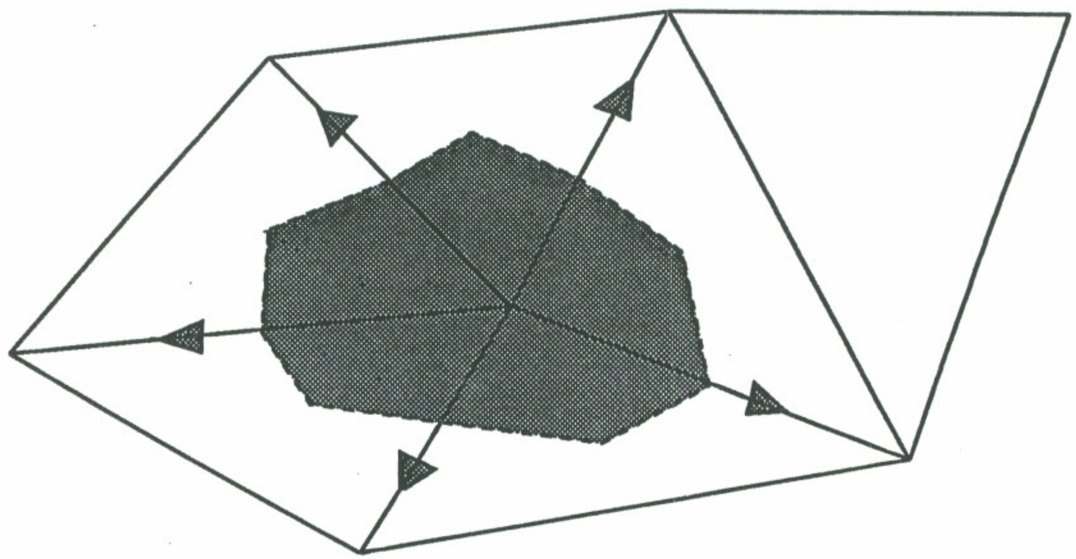
THE PROBLEM

- Twodimensional Euler solver on unstructured grid
- Vertex based finite volume scheme
- Piecewise linear reconstruction with limiters
- Fluxes computed using an approximate Riemann solver
- Time marching with fourth-order Runge-Kutta scheme
- Fluxes, gradients, and control volumes computed by looping over edges
- Vectorized code runs at 150 MFLOPS on one processor of Cray Y-MP

Unstructured Grid, Four Element Airfoil

- grid created by D. Jesperson, NASA Ames
- 15606 vertices
- 45878 edges
- 2 dimensional grid, four "holes"

Discretization



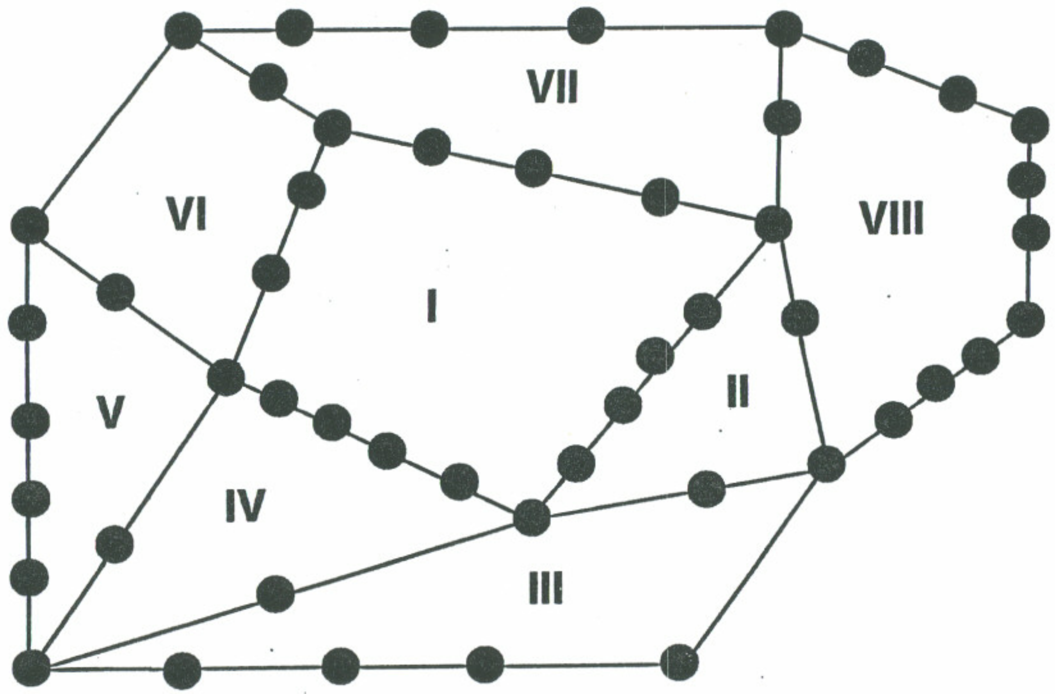
$$\frac{\partial \vec{W}}{\partial t} + \nabla \cdot \vec{F} = 0$$

$$\nabla \cdot \vec{F} = \frac{1}{2 A_c} \sum_{i=1}^{i=deg} [\, \vec{F}(u_i^-) + \vec{F}(u_i^+) - |\tilde{A}| \, (u_i^+ - u_i^-)]$$

Communication Issues

- Residuals,volumes, and gradient formed as integrals
- Partitions abut with no overlap even for higher order schemes
- Summing of local contributions to integrals
- Sparse matrix like data structures employed

Data Structures



nadj_proc = 7

iadj_proc = II,III,IV,V,VI,VII,VIII

ibv = 1,7,8,14,15,19,24,25

Dual Grid, Four Element Airfoil

- finite volume scheme, assign triangles to processors
- equivalently solve partitioning problem on the dual grid
- 30269 vertices (= triangles in original grid)
- 44929 edges (=interior edges in original grid)
- 2 dimensional grid, four "holes"

Communication Grid, Four Element Airfoil

- start with dual grid
- add edges corresponding to all messages sent
- 30269 vertices (= triangles in original grid)
- 178633 edges (=interior edges in original grid)
- all control volumes correspond to a clique in the graph

THREE PARTITIONING ALGORITHMS

General idea: find optimal strategy to partition domain into two subdomains, then apply recursively.

- recursive coordinate bisection
- recursive grid bisection (related to nested disection)
- recursive spectral bisection

All three only differ in the partitioning strategy for a simple domain into two subdomains.

Comparison of the three algorithms

- coordinate bisection creates long, narrow, and disconnected domains
- grid bisection creates more compact domains, but "fuzzy", and sometimes disconnected
- spectral bisection creates well balanced, connected domains
- spectral bisection is visually most pleasing

Performance of coordinate, grid, and spectral bisection

Table 1: Number of edges cut $|E_c|$ on airfoil problem

partitions	coordinate	grid	spectral
2	118	175	91
4	296	436	208
8	529	618	299
16	863	950	462
32	1193	1334	743
64	1653	1878	1154
128	2218	2529	1763

Intel iPSC/860 (Touchstone Gamma) at NAS

- 128 nodes based on the Intel i860 microprocessor (40 MHz)
- 8 Mbytes DRAM per nodes, 1 Gbyte total
- seven dimensional hypercube interconnect using iPSC/2 technology
- 10 I/O nodes based on 80386 processor
- 700 Mbytes disk storage per I/O node, 7 Gbytes total
- 80386 based host processor (SRM), running UNIX system 5
- About 7 GFLOPS advertised peak speed

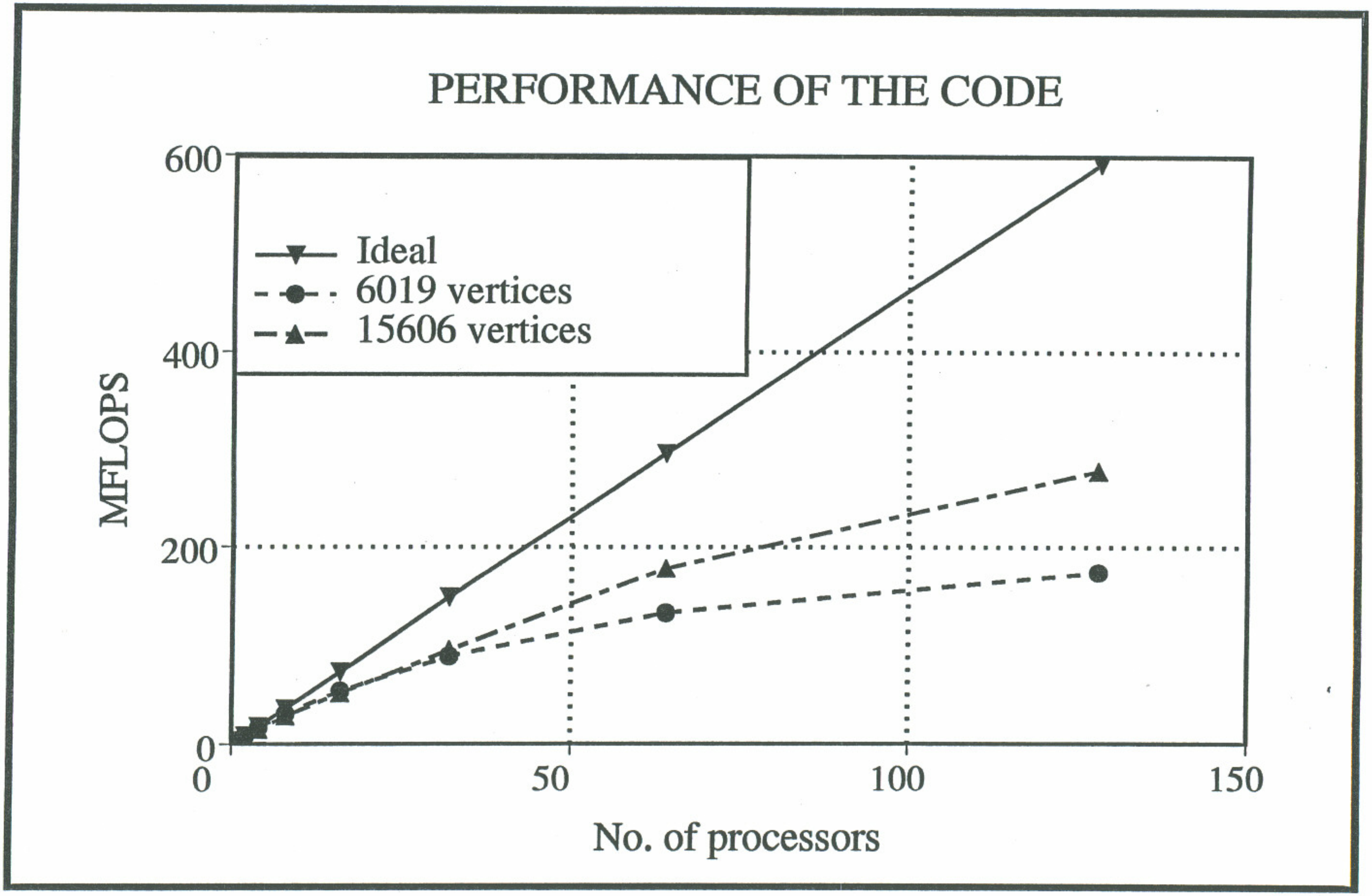
Performance of Unstructured Grid Euler Code on iPSC/860

Problem size (Vertices)		Intel								Cray-YM
		1	2	4	8	16	32	64	128	1
6019	Secs./step	4.86	3.05	1.31	0.71	0.41	0.25	0.17	0.13	0.15
	MFLOPS	4.63	7.4	17.1	31.6	54.9	90.0	132.4	173.1	150.0
	Efficiency(%)	100	80	93	86	74	61	45	29	-
15606	Secs./step	-	7.58	3.82	2.01	1.11	0.61	0.33	0.21	0.39
	MFLOPS	-	7.72	15.3	29.1	52.7	95.9	177.3	278.6	150.0
	Efficiency(%)	-	83	83	79	71	65	60	47	-

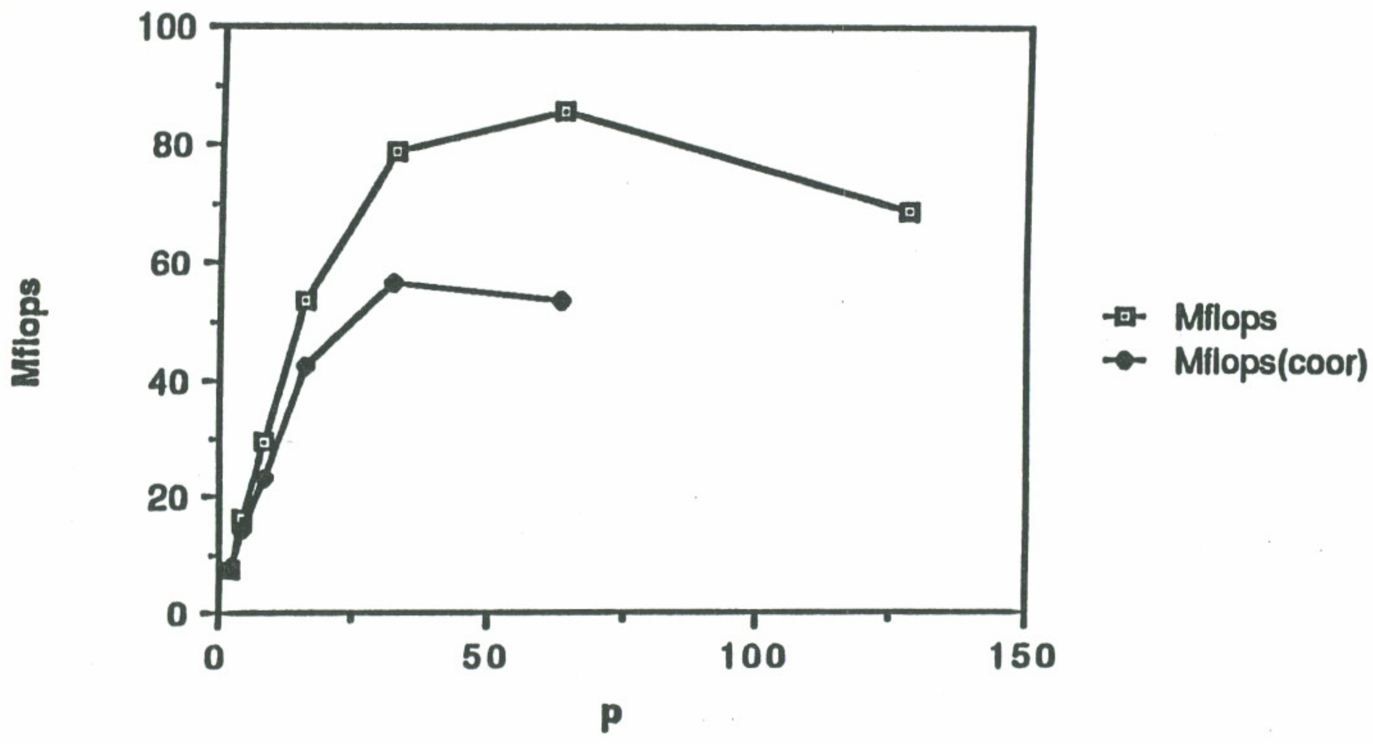
MFLOPS based on operation counts using the Cray hpm.

Efficiency(%) = (MFLOPS with N procs) / [N x (MFLOPS with 1 proc)] x 100

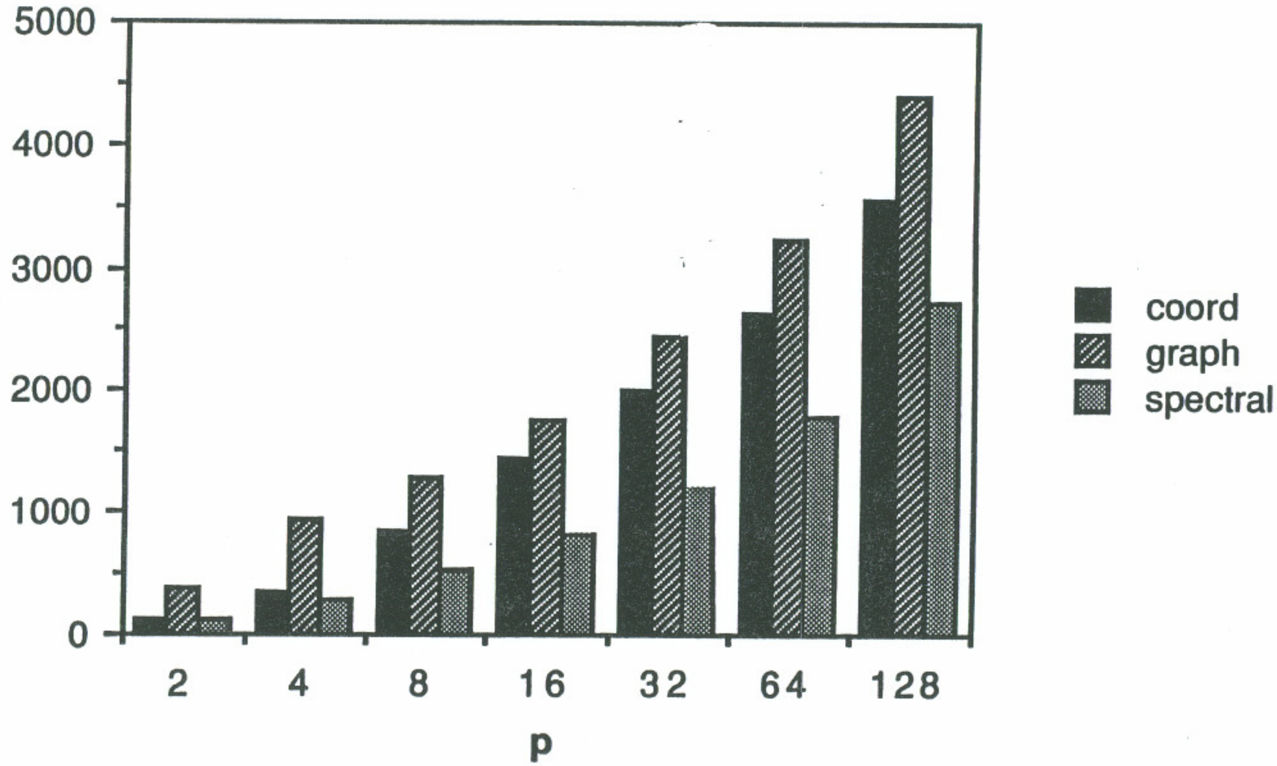
Performance of Unstructured Grid Euler Code on iPSC/860



Comparison of Different Partitioning Algorithms



Comparison of Different Partitioning Algorithms (cont.)



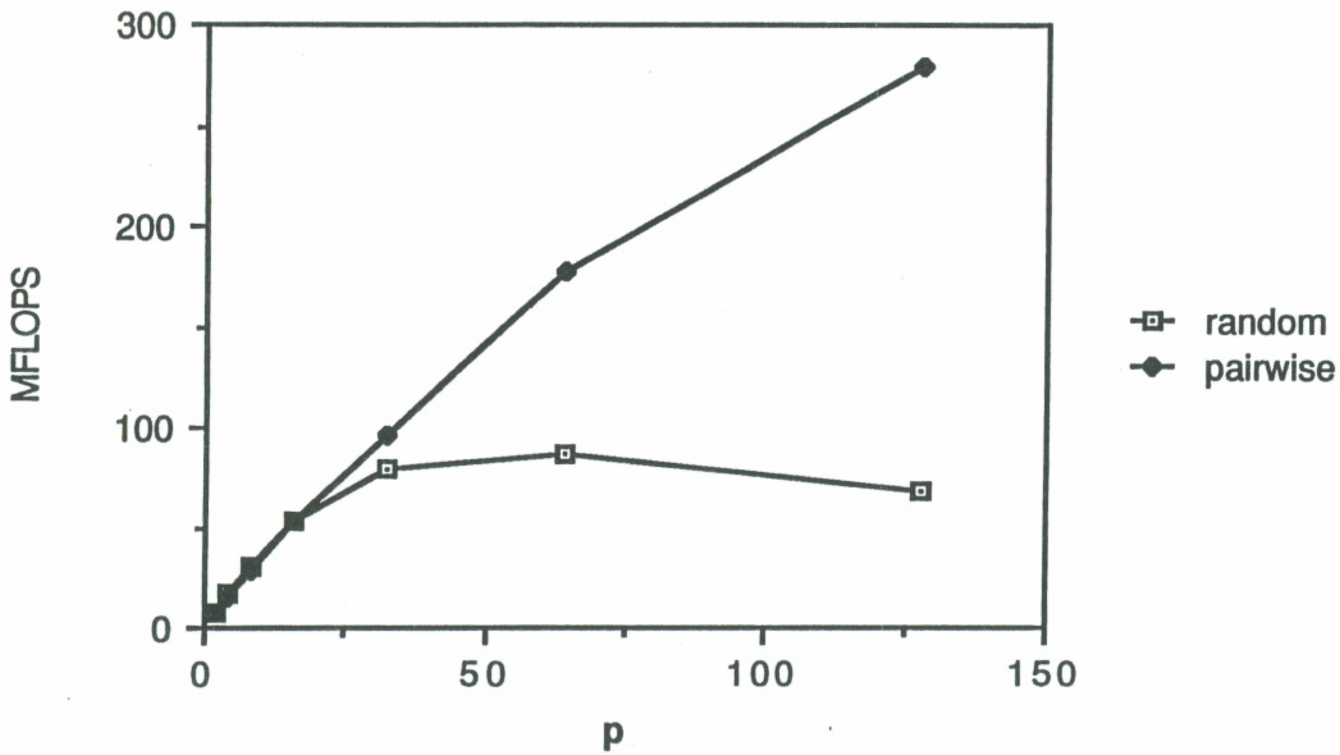
Comparison of Different Partitioning Algorithms (cont.)

PROBLEM SIZE : 15606 VERTICES

METHOD	SECS. (MFLOPS)	COMM. TIME	AVG. # NBRS	INT. BDY. VERTICES	MAX(NBRS,VERTIC
SPECTRAL	0.33 (177)	0.096	4.7	1819	12, 101
COORDINATE	0.41 (143)	0.173	6.7	2631	14, 120
COMM. GRAPH (SPECTRAL)	0.33 (177)	0.094	4.5	1791	14, 109

Message Passing Optimizations I

Permute adjacency processor list to achieve pairwise exachanges. Almost a factor of two improvement in performance.



Message Passing Optimizations II

Use stripwise decompositions in order to reduce the number of neighboring domains (message startup costs). No improvement, because of increased message lengths.

METHOD	SECS. (MFLOPS)	COMM. TIME	AVG. # NBRS	INT. BDY. VERTICES	MAX(NBRS,VERTICI
SPECTRAL	0.33 (177)	0.096	4.7	1819	12, 101
COORDINATE	0.41 (143)	0.173	6.7	2631	14, 120
COMM. GRAPH (SPECTRAL)	0.33 (177)	0.084	4.5	1791	14, 109
COMM. GRAPH (STRIPS)	0.55 (106)	0.310	3.7	6332	8, 382
COMM. GRAPH (RCM)	0.78 (75)	0.324	2.7	12406	4, 516

Message Passing Optimizations III

- Use pairwise forced exchange with synchronized messages.
- Suggested by Bokhari
- Obtained another 15% improvement on 64 processor run

SUMMARY

- implemented explicit flow solver on MIMD machine
- obtained close to 300 MFLOPS performance
- succesfully used new partitioning strategy
- careful implementation of message passing is critical, even for explicit code